**Lektion 18: Datenbanken und Tabellen erstellen**

In der vergangenen Lektion haben wir uns angeschaut, wie man eine Verbindung zum Datenbankserver herstellt. Der nächste Schritt ist jetzt, Datenbanken und Tabellen zu erstellen.

Wir schauen uns zwei Arten an, wie Datenbanken und Tabellen erstellt werden können. Zuerst, wie man es mit PHP machen kann und dann, wie dies mit der benutzerfreundlichen Software PhpMyAdmin, welche bei vielen Webhostern und in XAMPP Standard ist, gemacht wird.

Wenn Sie eine gehostete Webseite mit PHP und MySQL haben, dann wurde die Datenbank für Sie vielleicht schon eingerichtet. Dann können Sie diesen Teil überspringen und mit dem "Tabellen erstellen" weitermachen. Auch hier wieder der Hinweis: Informieren Sie sich bitte bei Ihrem Webhoster.

**Datenbanken und Tabellen in PHP erstellen**

Die Function [[Dokumentation](http://php.net/manual/en/function.mysql-query.php)mysql\_query](http://php.net/manual/en/function.mysql-query.php) wird benutzt, um eine Anfrage (query) an eine MySQL-Datenbank zu senden. Die Anfragen sind in der 'Sprache' **S**tructured **Q**uery **L**anguage (SQL) geschrieben. SQL ist die meist verbreitete Sprache für Datenbankabfragen - nicht nur für MySQL-Datenbanken. Sie ist sehr logisch und einfach zu lernen. In dieser und der nächsten Lektion, werden Sie ein paar der wichtigsten SQL-Abfragen kennenlernen.

Wenn Sie eine Datenbank erstellen, verwenden Sie die SQL-Query [[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/create-database.html)CREATE DATABASE](http://dev.mysql.com/doc/refman/5.5/en/create-database.html) mit der folgenden Syntax:

CREATE DATABASE *Datenbankname*

Logisch und einfach oder!? Lassen Sie und dieses in ein PHP-Skript schreiben:

mysql\_connect("mysql.meinserver.de", "benutzer001", "m31np455w0rt") or die(mysql\_error());

mysql\_query("CREATE DATABASE meinedatenbank") or die(mysql\_error());

mysql\_close();

Zuerst verbinden wir uns mit dem MySQL-Server. Dann erstellen wir eine Datenbank mit dem Namen "meinedatenbank" und zum Schluss trennen wir die Verbindung zum MySQL-Server wieder.

So weit, so gut ... die Dinge werden aber ein wenig komplizierter, wenn wir Tabellen mit PHP erstellen wollen. Dann nutzen wir die SQL-Query [[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/create-table.html)CREATE TABLE](http://dev.mysql.com/doc/refman/5.5/en/create-table.html) mit der folgenden Syntax:

CREATE TABLE *tabellenname*

(

*spalte1\_name DATENTYP,*

*spalte2\_name DATENTYP,*

*spalte3\_name DATENTYP,*

...

)

*tabellenname* und *spalteX\_name* sind natürlich der Name der Tabelle und der Spalten. *DATENTYP* wird dazu genutzt, um den Datentyp der einzugebenden Daten festzulegen. Die meistgenutzten Datentypen sind:

[[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/numeric-types.html)INT](http://dev.mysql.com/doc/refman/5.5/en/numeric-types.html)

Für ganze Zahlen (ohne Dezimalstellen)

[[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/numeric-types.html)DECIMAL](http://dev.mysql.com/doc/refman/5.5/en/numeric-types.html)

Für Zahlen mit Dezimalstellen

[[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/char.html)CHAR](http://dev.mysql.com/doc/refman/5.5/en/char.html)

Kurzer Text mit bis zu 255 Zeichen

[[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/blob.html)TEXT](http://dev.mysql.com/doc/refman/5.5/en/blob.html)

Für Text mit bis zu 65.535 Zeichen

[[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/blob.html)LONGTEXT](http://dev.mysql.com/doc/refman/5.5/en/blob.html)

Für lange Textpassagen mit bis zu 4.294.967.295 Zeichen

[[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/datetime.html)Date](http://dev.mysql.com/doc/refman/5.5/en/datetime.html)

Für Daten im Format JJJJ-MM-TT

[[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/time.html)Time](http://dev.mysql.com/doc/refman/5.5/en/time.html)

Für Zeitangaben im Format HH:MM:SS

[[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/datetime.html)DATETIME](http://dev.mysql.com/doc/refman/5.5/en/datetime.html)

Für Datums- und Zeitangaben im Format JJJJ-MM-TT HH:MM:SS

Alles in allem, logisch und relativ einfach. Lassen Sie uns dies in einem Beispiel anschauen:

mysql\_connect("mysql.meinserver.de", "benutzer001", "m31np455w0rt") or die(mysql\_error());

mysql\_select\_db("Menschen") or die(mysql\_error());

mysql\_query("CREATE TABLE Personen (

id INT AUTO\_INCREMENT,

Vorname CHAR,

Name CHAR,

Telefon INT,

Geburtsdatum DATE

PRIMARY KEY(id)

)") Or die(mysql\_error());

mysql\_close ();

Im Beispiel verbinden wir uns wieder zuerst mit dem MySQL-Server. Dann verwenden wir die Function [[Dokumentation](http://php.net/manual/en/function.mysql-select-db.php)mysql\_select\_db](http://php.net/manual/en/function.mysql-select-db.php) um die Datenbank "Menschen" auszuwählen. Danach erstellen wir die Tabelle "Personen" mit fünf Spalten.

Beachten Sie bitte, dass wir bei der "id"-Spalte zuerst [[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/numeric-types.html)INT](http://dev.mysql.com/doc/refman/5.5/en/numeric-types.html) zuweisen, um festzulegen, dass die Spalte Zahlen enthält und dann [[Dokumentation](http://dev.mysql.com/doc/refman/5.0/en/example-auto-increment.html)AUTO\_INCREMENT](http://dev.mysql.com/doc/refman/5.0/en/example-auto-increment.html), damit sich diese Zahl automatisch um 1 erhöht, um sicherzustellen, dass jede Zeile eine eindeutige ID hat.

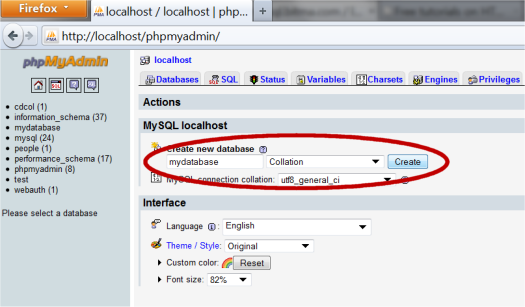
Zum Schluss benutzen wir [[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/constraint-primary-key.html)PRIMARY KEY](http://dev.mysql.com/doc/refman/5.5/en/constraint-primary-key.html) dafür, die "id"-Spalte als Primärschlüssel (primary key) festzulegen. Der Primärschlüssel identifiziert jeden Eintrag (=Zeile) in der Tabelle, was später, wenn wir die Datenbank aktualisieren, wichtig und nützlich ist.

**Datenbanken und Tabellen mit phpMyAdmin erstellen**

Es kann sehr nützlich sein, in der Lage zu sein Datenbanken und Tabellen direkt in PHP erstellen zu können. Oftmals ist es aber einfacher phpMyAdmin (oder auch eine beliebige andere MySQL-Administrations-Software), die Standard bei vielen Hostern und XAMPP ist, zu benutzen. Die nachfolgenden Screenshots zeigen, wie man eine Datenbank und Tabellen in phpMyAdmin anlegt.

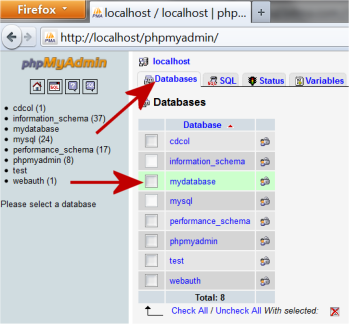
Am Anfang müssen Sie sich bei phpMyAdmin einloggen. Oft ist die Adresse die gleiche, wie die Ihres MySQL-Servers (z.B. "http://mysql.meinserver.de") und Sie können auch die gleiche Benutzername-Passwort-Kombination verwenden. In XAMPP ist die Adresse http://localhost/phpmyadmin/.

Wenn Sie angemeldet sind, tippen Sie einfach den Namen, den Sie der Datenbank geben möchten, ein und klicken auf "Erstellen" (bzw. "Create"):

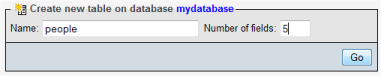


Bei manchen Hostern ist es möglich, dass bereits eine Datenbank vorkonfiguriert ist und Sie nicht die Rechte haben, (eine) weitere zu erstellen. Wenn dies der Fall ist, können Sie offensichtlich nur die vorgegebene Datenbank nutzen.

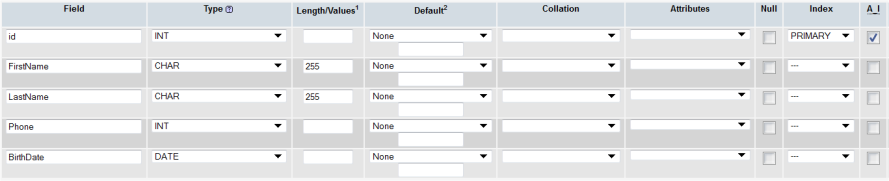
Um eine Tabelle anzulegen, klicken Sie auf den Reiter "Datenbanken" ("Databases") und wählen eine Datenbank aus, indem Sie darauf klicken:



Dann erscheint ein Fenster mit dem Titel "Neue Tabelle in Datenbank erstellen", wo Sie den Namen der Tabelle eingeben und die Anzahl der Spalten festlegen können. Drücken Sie dann auf den Button "Los" ("Go"):



Dann können Sie die Spalten benennen und den Datentyp usw., wie im PHP/SQL-Beispiel oben, festlegen.



Beachten Sie bitte, dass wir auch hier "id" als [[Dokumentation](http://dev.mysql.com/doc/refman/5.5/en/constraint-primary-key.html)PRIMARY KEY](http://dev.mysql.com/doc/refman/5.5/en/constraint-primary-key.html) festlegen und [[Dokumentation](http://dev.mysql.com/doc/refman/5.0/en/example-auto-increment.html)AUTO\_INCREMENT](http://dev.mysql.com/doc/refman/5.0/en/example-auto-increment.html) (A\_I) benutzen.

Jetzt haben Sie ihre eigene Datenbank und Tabelle erstellt. Weiter geht es in der nächsten Lektion mit dem Einfügen, Auffinden/Ausgeben und Löschen von Daten in/aus einer Datenbank.